**BAB III**

**LANDASAN TEORI**

## Internet dan Aplikasi Web

Menurut Jogiyanto (2004, h.20) dalam bukunya yang berjudul “Analisis dan Desain Sistem Informasi” aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Jogiyanto menambahkan aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal.

Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan *software* yang ditransformasikan ke komputer yang berisikan perintah-perintah yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

### Sekilas Tentang *Internet*

Menurut (Iskandar, 2009) dalam buku Panduan Lengkap *Internet*, Jaringan *internet* pertama kali dikembangkan pada 1969 oleh Departemen Pertahanan Amerika Serikat dengan nama *ARPANET (US Defense Adavanced Research Projects Agency). ARPANET* dibangun untuk menghindari pemusatan informasi di satu titik dengan pembuatan suatu jaringan komputer yang tersebar.

Tahun 1980-an, *ARPANET* terpecah menjadi dua jaringan, yaitu *ARPANET* dan *Milnet* (sebuah jaringan militer). Akan tetapi, keduanya mempunya hubungan sehingga komunikasi antarjaringan tetap dapat dilakukan.

Jaringan interkoneksi mulanya disebut *DARPA Internet.* Sesudah itu, *internet* mulai digunakan untuk kepentingan akademis. Kemudian, menyusul dibukanya layanan *Usenet* dan *Bitnet, internet* mulai dapat diakses melalui komputer pribadi (*PC*). Selanjutnya, pada 1982 protokol standar *TCP/IP* mulai diperkenalkan, disusul dengan penggunaan sistem *DNS (Domain Name Services)* pada tahun 1984.

Tahun 1986, lahir *National Science Foundation Network (NSFNET)* yang menghubungkan para periset di seluruh negeri dengan lima buah pusat superkomputer. *NSFNET* kemudian mulai menggatikan *ARPANET*sebagai jaringan riset utama di Amerika. Pada bulam Maret 1990 *ARPANET* secara resmi dibubarkan.

Terobosan lainnya terjadi pada 1993 ketika *InterNIC* didirikan untuk menjalankan layanan pendaftaran *domain*. Bersamaan dengan itu*, White House* mulai *online* di *internet* dan pemerintah Amerika Serikat meloloskan *National Information Infrastructure Act*. Penggunaan *internet* secara komersil dimulai pada 1994 dipelopori oleh perushaan *Pizza* *Hut*, sedangkan *internet banking* pertama kali diaplikasikan oleh *First Virtual*. Setahun kemudian, *compurserve*, *America Online*, dan *Prodigy* mulai memberikan layanan akses ke *internet* bagi masyarakat umum.

Masyarakat Indonesia baru bisa menikmati layanan *internet* komersil sekitar tahun 1994. Sebelumnya, beberapa perguruan tinggi seperti Universitas Indonesia telah terlebih dahulu tersambung dengan jaringan *internet* melalui *gateway* yang menghubungkan universitas itu dengan *network* luar negeri.

### Aplikasi *Internet* dan Web

Menurut Iskandar (2009, h.59) dalam bukunya yang berjudul “Panduan Lengkap *Internet*", *Internet atau interconnected network* adalah sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia. Setiap komputer dan jaringan terhubung secara langsung maupun tidak langsung ke beberapa jalur utama yang disebut *internet* *backbone*. Masing-masing dibedakan antara satu dengan yang lainnya menggunakan *unique* name yang disebut alamat IP 32 bit.

Komputer dan jaringan dengan berbagai *platform* (*Unix, Linux, Windows, Mac* dan lain-lain) dapat bertukar informasi dengan adanya sebuah protokol standar yang dikenal dengan nama *TCP*/*IP* (*Transmission Control Protocol/Internet Protocol*). *TCP*/*IP* tersusun atas 4 layer, yaitu *network access, internet, host-to-host transport,* dan *application*.

*Internet* merupakan sekumpulan jaringan komputer yang menghubungkan situs akademik, pemerintahan, komersial, organisasi, maupun perorangan. *Internet* menyediakan akses untuk layanan telekomunikasi dan sumber daya informasi untuk jutaan pemakainya yang tersebar di seluruh dunia. Layanan *internet* meliputi komunikasi langsung (*email* atau *chat*), diskusi (*Usenet news, email, milis*), sumber daya informasi yang terdistribusi (*World Wide web, Gopher*), *remote login*, lalu lintas *file* (*Telnet ,FTP*), dan aneka layanan lainnya.

#### Aplikasi *Internet*

Menurut (Iskandar, 2009) dalam buku Panduan Lengkap *Internet, Interne*t memiliki aplikasi yang paling sering digunakan oleh pengguna *internet*, antara lain :

1. *WWW (World Wide Web)*

*WWW* atau *web* merupakan aplikasi *internet* yang paling populer. Secara teknis, *web* merupakan sebuah sistem dengan informasi dalam bentuk teks, gambar, suara dan lain-lain yang tersimpan dalam sebuah *internet webserver* dan dipresentasikan dalam bentuk *hypertext*. Informasi dalam *web* yang berupa teks umumnya ditulis dalam format *HTML (Hypertext Markup Language)*. Informasi lainnya disajikan dalam bentuk grafis (dalam format *GIF, JPG, PNG*) suara (dalam bentuk *AU, WAV*) dan objek multimedia lainnya (seperti *MIDI, Shockwave, Quicktime Movie, 3D World*).

1. *E-mail*

*Email* adalah aplikasi yang memungkinkan para pengguna *internet* saling berkirim pesan melalui alamat eletronik di *internet*. Para pengguna *email* memiliki sebuah *mailbox* elektronik yang tersimpan dalam *mailserver*. Suatu *mailbox* memiliki sebuah alamat sebagai pengenal agar dapat berhubungan dengan *mailbox* lainnya, baik dalam bentuk penerimaan maupun pengiriman pesan, menghapus, atau menyunting dan mengirimkan pesan *email*.

1. *File transfer*

Fasilitas ini memungkinkan para pengguna *internet* untuk melakukan pengiriman (*upload*) atau penyalinan (*download*) *file* antar komputer lokal denngan komputer lain yang terhubung dalam jaringan *internet*. Protokol standar yang digunakan untuk keperluan ini disebut *File Transfer Protocol (FTP).*

*FTP* umunya dimanfaatkan sebagai sarana pendukung untuk kepentingan pertukaran penyebarluasan file melaui jaringa *internet. FTP* juga dimanfaatkan untuk mengunggah (meng-*upload*) halaman *web* ke *webserver* agar halaman *web* tersebut dapat diakses oleh pengguna *internet* lainnya.

1. *Remote Login*

Layanan *remote login* mengacu pada program atau protokol yang menyediakan fungsi yang memungkinkan pengguna *internet* untuk mengakses (*login*) ke sebuah terminal (*remote host*) dalam lingkungan jaringan *internet*. Dengan memanfaatkan *remote login*, pengguna *internet* dapat mengoperasikan sebuah *host* bersangkutan. Dari sana dapat melakukan pemeliharaan (*maintenance*), menjalankan sebuah program, atau dapat menginstal program baru di *remote host.*

#### Pemrograman Aplikasi *Web*

1. *HTML*

Menurut (Abdul Kadir, 2011) dalam Buku Pintar *jQuery* dan *PHP*, saat ini banyak sekali bahasa pemrograman untuk *web*. Namun, *HTML* tetap menjadi tulang pungggung aplikasi *web*. Sekalipun sejumlah teknologi pendukung aplikasi *web* telah muncul, peran *HTML* tetap tidak bisa tergantikan. Kehadiran seperti *PHP* atau *ASP* merupakan sebagai pelengkap yang memungkinkan aplikasi *web* tidak lagi bersifat statis, melainkan bersifat dinamis. Perubahan-perubahan pada sisi data tidak perlu membuat aplikasi diubah. Cara seperti itu terutama dilakukan pada sisi *sever*. Artinya, kode seperti *PHP* dan *ASP* berjalan di *server*. Selain cara seperti itu, penambhan kode pada *HTML* juga bisa dilakukan pada sisi klien.

1. *CSS*

*Cascading Style Sheet* (CSS) merupakan salah satu bahasa pemrograman *web* untuk mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam. Sama halnya *styles* dalam aplikasi pengolahan kata seperti *Microsoft* *Word* yang dapat mengatur beberapa *style*, misalnya *heading*, *subbab*, *bodytext*, *footer*, *images*, dan *style* lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (*file*). Pada umumnya *CSS* dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa *HTML* dan *XHTML*.

*CSS* dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran *border*, warna *border*, warna *hyperlink*, warna *mouse over*, spasi antar paragraf, spasi antar teks, *margin* kiri, kanan, atas, bawah, dan parameter lainnya. *CSS* adalah bahasa *style sheet* yang digunakan untuk mengatur tampilan dokumen. Dengan adanya *CSS* memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

Ada dua sifat *CSS* yaitu *internal* dan *eksternal*. Jika *internal* yang dipilih, maka skrip itu dimasukkan secara langsung ke halaman *website* yang akan didesain. Kalau halaman *web* yang lain akan didesain dengan model yang sama, maka skrip *CSS* itu harus dimasukkan lagi ke dalam halaman *web* yang lain itu.

Sifat yang kedua adalah *eksternal* di mana skrip *CSS* dipisahkan dan diletakkan dalam berkas khusus. Nanti, cukup gunakan semacam tautan menuju berkas *CSS* itu jika halaman *web* yang didesain akan dibuat seperti model yang ada di skrip tersebut. (Abdul Kadir, 2011).

1. *PHP*

Menurut (Abdul Kadir, 2011) dala Buku Pintar *Jquery* dan *PHP*, *PHP* dikenal sebagai bahasa pemrograman yang kodenya dijalankan disisi *server*. Dengan demikian kode aslinya tidak akan terlihat klien (*browser*). *PHP* banyak dipakai dalam membuat aplikasi *web* antara lain karena dukungan pustaka yang lengkap dan mudah digunakan pada berbagai *platform* (*Windows, Mac, Linux*, dan lain-lain). Dengan menggunakan *PHP*, koneksi ke *database server* (penyedia daya) juga mudah dilakukan.

*PHP* sendiri adalah perngkat lunak yang bersifat gratis. Namun, perlu diketahui, *PHP* terkadang dikemas dalam bundel perngkat lunak, misalnya pada ***WampServer***. Hal yang menarik lainnya, *PHP* bersifat *multiplatform*. Artinya, *PHP* dapat berjalan pada berbagai sistem, seperti *Windows, Linux,* dan *UNIX.*

1. *jQuery*

Menurut (Lukmanul Hakim, 2010) dalam buku Bikin *Website* Super Keren dengan *PHP & jQuery, jQuery* adalah *Javascipt Library* berupa kumpulan kode/fungsi *javascript* siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode *javascript.* Secara standar, apabila kita membuat kode *javascipt,* maka diperlukan kode yang cukup panjang, bahkan terkadang sangat sulit dipahami, *jQuery* digunakan untuk menyederhanakan kode *javascript*. Hal ini sesuai dengan slogannya *“Write lees, do more*”, cukup tulis sedikit tapi bisa melakukan banyak hal.

1. *Javascipt*

*Javascript* adalah bahasa *scripting* yang populer di *internet* dan dapat bekerja di sebagian besar browser popular seperti *Internet Exploler, Mozzila Firefox, Netscape* dan *Opera*. Kode *javascript* dapat disisipkan dalam halaman *web* menggunakan *tag* *script*.

Beberapa hal tentang *JavaScript* sebagai berikut:

1. *JavaScript* didesain untuk menambah interaktif suatu *web*.
2. *JavaScript* merupakan sebuah bahasa *scripting*.
3. Bahasa *scripting* merupakan bahasa pemrograman yang ringan.
4. *JavaScript* berisi baris kode yang dijalankan di komputer (*web browser*).
5. *JavaScript* biasanya disisipkan (*embedded*) dalam halaman *HTML.*
6. *JavaScript* adalah bahasa *interpreter* (yang berarti skrip dieksekusi tanpa   
    proses kompilasi.
7. *Database Server : MySQL*

*MYSQL* adalah *database*. *Database* sendiri merupakan suatu jalan untuk dapat menyimpan berbagai informasi dengan membaginya berdasarkan kategori-kategori tertentu. Dimana informasi-informasi tersebut saling berkaitan satu dengan yang lainnya.

*MYSQL* bersifat *RDBMS* (*Relational Database Management System*), yang memungkinkan seorang admin dapat menyimpan banyak informasi ke dalam tabel-tabel, dimana tabel-tabel tersebut saling berkaitan satu sama lain. Keuntungan *RDBMS* sendiri adalah kita dapat memecah *database* kedalam tabel-tabel yang berbeda. Setiap tabel memiliki informasi yang saling berkaitan dengan tabel yang lainnya.

Sama dengan *PHP*, *MYSQL* bersifat *opensource*, yang artinya semua orang dapat menggunakannya dengan gratis. *MYSQL* juga bersifat *Cross Platform,* yang artinya dapat digunakan *under* *Windows* ataupun *under* *Linux*. (Abdul Kadir, 2011).

1. *Bootstrap*

*Bootstrap* adalah sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain halaman *website*. Sesuai namanya, *website* yang dibuat dengan alat bantu ini memiliki tampilan halaman yang sama/mirip dengan tampilan halaman *Twitter* atau desainer juga dapat mengubah tampilan halaman *website* sesuai dengan kebutuhan.

*Bootstrap* dibangun dengan teknologi *HTML* dan *CSS* yang dapat membuat *layout* halaman *website*, tabel, tombol*, form*, navigasi, dan komponen lainnya dalam sebuah *website* hanya dengan memanggil fungsi *CSS (class)* dalam berkas *HTML* yang telah didefinisikan. Selain itu juga terdapat komponen-komponen lainnya yang dibangun menggunakan *JavaScript.*

## Pemodelan Sistem dengan UML

### UML (Unified Modeling Language)

*UML* singkatan dari *Unified Modelling Language* yang berarti bahasa pemodelan standar. *UML* memiliki sintakas dan semantik. Ketika kita membuat model menggunakan konsep *UML* ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan kontekasnya.

*UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya (Prabowo Pudjo Widodo, 2011)

### Use Case Diagram

1. Tabel Komponen *Use case*

Diagram *use case* menyajikan interaksi antara *use case* dan aktor. Dimana aktor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai. Beberapa komponen dalam *use case* antara lain

**Tabel 3.1** Komponen *Use Case Diagram*

|  |  |
| --- | --- |
| Komponen *Use* *case* | Penjelasan |
| 1.jpg  Aktor | Merupakan sebuah komponen yang menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lainnya) yang berinteraksi dengan sistem. |
| *2.jpgUse Case* | *Use case* adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. |

Ada beberapa relasi yang terdapat dalam *use case*, antaran lain :

**Tabel 3.2** Relasi *Use Case Diagram*

|  |  |
| --- | --- |
| Relasi *Use case* | Penjelasan |
| Relasi *Association* | *Association*, menghubungkan *link* antar elemen |
| 3.jpg    Relasi *Generalization* | *Generalization* disebut juga *inheritance* (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya. |
| Relasi *Depedency* | *Dependency*, sebuah elemen bergantung dalam beberapa cara ke elemen lainnya. |

Tipe relasi / *stereotype* yang mungkin terjadi pada *use case diagram* :

**Tabel 3.3** Tabel *Stereotype* yang mungkin terjadi

|  |  |
| --- | --- |
| Relasi/ *Stereotype* | Penjelasan |
| **<<*include*>>** | Kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya. |
| **<<*extends*>>** | Kelakuan yang hanya berjalan dibawah kondisi tertentu seperti menggerakan alarm. |
| ***<<Communicates>>*** | Ditambahkan untuk asosiasi yang mungkin menunjukan asosiasinya adalah *communicates association*. Ini merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara *actor* dan *use case*. |

### *Activity* *Diagram*

Diagram aktivitas menggambarkan aliran fungsionalitas sistem. Dapat juga digunakan untuk menggambarkan aliran kejadian dalam *use case*. Berikut beberapa komponen yang terdapat dalam *Activity Diagram* :

**Tabel 3.4** *Activity Diagram*

|  |  |
| --- | --- |
| *Activity Diagram* | Penjelasan |
| 4.jpg  *Start State* | *Start State*, Sebagai tanda awal proses dari *activity diagram*. |
| 4.jpg  *State* | *State*, Berfungsi menampung *event* dalam *activity diagram.* |
| 5.jpg  *Activity* | *Activity*, Memiliki fungsi yang sama dengan *state*. Menampung *event* atau aktifitas pada proses sistem. |
| *State Transition* | *State Transition*, Berfungsi untuk menunjukan aliran atau urutan dari *event* atau aktifitas pada *diagram*. |
| 8.jpg  *Transition to self* | *Transition to Self*, Berfungsi untuk menunjukan transisi sebuah *event* yang mengarah ke *event* itu sendiri. |
| 3.jpg  *Horizontal Synchronization* | *Horizontal Synchronization*, Berfungsi untuk mengsinkronisasikan 2 cabang *event* yang posisinya *horizontal*. |
| 7.jpg  *Vertical Synchronization* | *Vertical Synchronization*, Berfungsi untuk meng*sinkron*isasikan 2 cabang *event* yang posisinya *vertikal*. |
| *6.jpgDecision* | *Decision*, Digunakan ketika terjadi pemilihan 2 kondisi *event* pada *diagram*. |
| 10.jpgEnd State | *End State*, Sebagai tanda akhir dari *activity diagram*. |

### *Squence* *Diagram*

Diagram sekuensial umumnya digunakan untuk menggambarkan suatu skenario atau urutan langkah-langkah yang dilakukan baik oleh *actor* maupun sistem yang merupakan respon dari sebuah kejadian untuk mendapatkan hasil atau *output*. Berikut adalah beberapa komponen yang terdapat dalam *sequence diagram* :

**Tabel 3.5** *Sequence Diagram*

|  |  |
| --- | --- |
| *Sequence Diagram* | Penjelasan |
| 1.jpg  *Actor* | *Actor*, Menggambarkan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem. |
| 1.jpg*Boxes* | *Boxes*, sebuah kotak yang tampil pada posisi paling atas diagram, yang mewakili object, use case, class, dan actor. |
| *Return Message* | *Return Message*, menggambarkan pesan atau hubungan antar obyek yang menunjukan urutan kejadian yang terjadi. |
| *Lifeline* | *Lifeline*, Eksekusi obyek selama *sequence* (*message* dikirim atau diterima dan aktifasinya) |
| *Message to Self* | *Message to Self*, Menggambarkan pesan atau hubungan obyek itu sendiri yang menunjukan urutan kejadian yang terjadi. |
| *Object Message* | *Object Message*, Menggambarkan pesan atau hubungan antar obyek yang menunjukan urutan kejadian yang terjadi. |